

IHM	
-hConsole	HANDLE
-activerCouleur	vector<string>
-colorerEcran(int code_couleur)	bool
+choixDepartArrivee(metro : M	void
+choixTypeItineraire()	bool
+afficherItineraire(itineraire : I	void
+quitter()	bool

Itineraire	
-arretsItineraire	vector<Arret*>
-tempsTotal	int
+Itineraire(destination : Node*, metr	Itineraire
+creerItineraireSimplifie()	vector<Arret*>
+creerNbArretsSimplifie()	vector<int>

Metro	
-arretsMetro	vector<Arret*>
-lignesMetro	vector<Ligne*>
+importerDonnees(wd : string)	void
+getArret(int idArret)	Arret*
+getLigne(int route_id)	Ligne*
-setArretsMetro(arret : Arret*)	void
-importerToutesLignesTxt(wd : void	
-importerTousStopTxt(wd : strir	void
-importerStopTxt(path_fichier : void	
-importerLigneTxt(path_fichier	bool

Ligne	
-idLigne	int
-numero_ligne	string
-nom_trajet	string
-couleur	string
-direction	string
-arretsLigne	vector<Arret*>
+setArretsLigne(arret : void	

Arret	
-idArret	int
-nom	string
-adresse	string
-stop_lon	float
-stop_lat	float
-lignesArret	vector<Ligne*>
+setLignesArret(ligne : Li	void
+memeLigne (arret : Arre	bool
+memeArret (arret : Arre	bool
-calculerDirection(arret : void	vector<string>

M tro

Graphe	
-nodes	vector<Node*>
-edges	vector<Edge*>
-adjacentRemainingNodes(node : Node*)	vector<Node*>*
-initialiserNodes()	void
-importerNodes(chemin : string)	void
-importerEdges(chemin : string, chemin_changement : strir	void
-extractSmallest(&nodes : vector<Node*>, minChangement	Node*
-contains(&nodes : vector<Node*>, node : Node*)	bool
-calculerDistance(node1 : Node* , node2 : Node*, minChan	int
+dijkstras(entree : string, sortie : string, minChangement : Node*	
-getIndiceFromNode(identifiant : string, les_noeuds : vecto	int

Edge	
+node1	Node*
+node2	Node*
-distanceCourtChemin	int
-distanceMinChangement	int
+connects(node1 : Node* , node2	bool
+getDistance(minChangement : b	int

Node	
-idNode	int
-distanceFromStartCourtChemin	int
-distanceFromStartMinChangement	int
+previous	Node*
+setDistanceFromStart(dist : int, minChangevoid	: bool)
+getDistanceFromStart(minChangement : b	int
+initialiserNode()	void

Algorithme